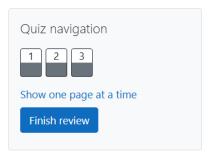
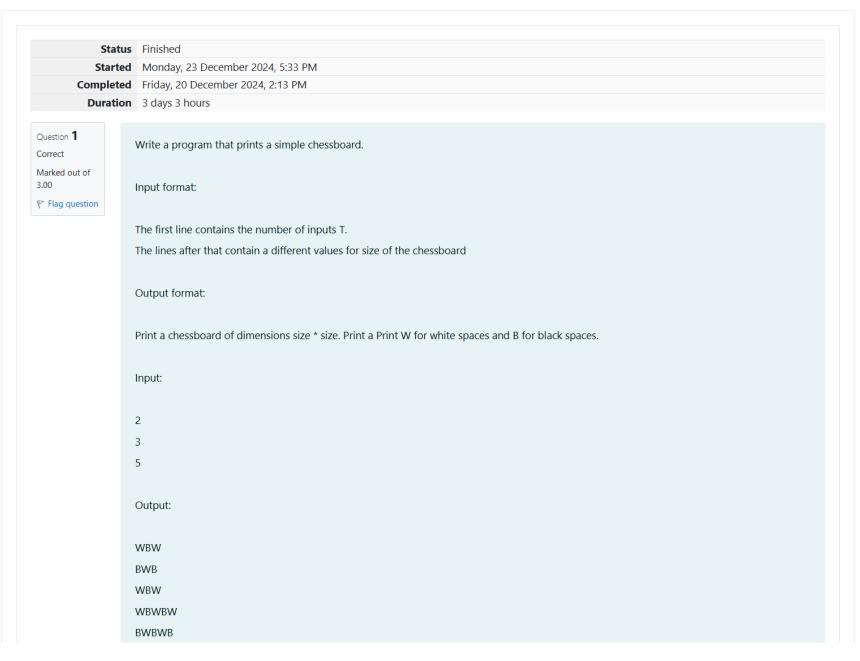
GE23131-Programming Using C-2024





WBWBW BWBWB WBWBW

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
 2
    int main()
 3 ₹ {
       int T,d,i=0,i1,i2,o;
 4
 5
        char c;
       scanf("%d",&T);
 6
       while(i<T)
 7
 8
           scanf("%d",&d);
 9
10
           i1=0;
           while(i1<d)
11
12
               o=1;
13
14
               i2=0;
               if(i1%2==0)
15
16
                  0=0;
17
18
19
               while(i2<d)
20
                  c='B';
21
                  if(i2%2==o)
22
23
                      c='W';
24
25
26
                  printf("%c",c);
27
28
                  i2++;
29
30
               i1+=1;
31
               printf("\n");
32
33
           i=i+1;
34
35
36 }
```

	Input	Expected	Got	
~	2	WBW	WBW	~
	3	BWB	BWB	
	5	WBW	WBW	
		WBWBW	WBWBW	
		RMRMR	BMBMB	

WBWBW WBWBW BWBWB WBWBW WBWBW WBWBW WBWBW

Question 2

Correct

Marked out of

Flag question

Let's print a chessboard! Write a program that takes input: The first line contains T, the number of test cases Each test case contains an integer N and also the starting character of the chessboard Output Format Print the chessboard as per the given examples Sample Input / Output Input: 2 2 W 3 B Output: WB BW **BWB**

Answer: (penalty regime: 0 %)

WBW BWB

```
#include<stdio.h>
    int main()
 2
 3 ₹ {
 4
        int T,d,i,i1,i2,o,z;
 5
        char c,s;
        scanf("%d",&T);
 6
        for(i=0;i<T;i++)
 7
 8
            scanf("%d %c",&d,&s);
 9
            for(i1=0;i1<d;i1++)</pre>
10
11 1
               z=(s=='W')?0:1;
12
13
                o=(i1%2==z)?0:1;
                for(i2=0;i2<d;i2++)
14
15
                   c=(i2%2==o)?'W':'B';
16
17
                   printf("%c",c);
18
               printf("\n");
19
20
21
22
        return 0;
23 }
```

/	2	WB	WB	~
	2 W	BW	BW	
	3 B	BWB	BWB	
		WBW	WBW	
		BWB	BWB	

Question **3**

Correct Marked out of 7.00

Flag question

Decode the logic and print the Pattern that corresponds to given input.

If N= 3

then pattern will be:

10203010011012

**4050809

****607

```
If N= 4, then pattern will be:
1020304017018019020
**50607014015016
****809012013
*****10011
Constraints
2 <= N <= 100
Input Format
First line contains T, the number of test cases
Each test case contains a single integer N
Output
First line print Case #i where i is the test case number
In the subsequent line, print the pattern
Test Case 1
3
3
4
5
Output
Case #1
10203010011012
**4050809
****607
```

```
Case #2
1020304017018019020
**50607014015016
****809012013
******10011
Case #3
102030405026027028029030
**6070809022023024025
****10011012019020021
******13014017018
*******15016
```

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
 2
    int main()
 3 ₹
 4
        int n,v,p3,c,in,i,i1,i2,t,ti;
 5
         scanf("%d",&t);
 6
         for(ti=0;ti<t;ti++){</pre>
 7
 8
            scanf("%d",&n);
 9
            printf("Case #%d\n",ti+1);
            for(i=0;i<n;i++){</pre>
10
11
                c=0;
12
                if(i>0){
13
                     for(i1=0;i1<i;i1++) printf("**");</pre>
14
15
                 for(i1=i;i1<n;i1++){</pre>
16
17
                    if(i>0) c++;
18
                    printf("%d0",++v);
19
20
                if(i==0){
21
                    p3=v+(v*(v-1))+1;
22
                    in=p3;
23
24
                in=in-c;
25
                p3=in;
26
                for(i2=i;i2<n;i2++){
27
                    printf("%d",p3++);
                    if(i2!=n-1) printf("0");
28
29
30
                }printf("\n");
31
32
33
34
```

	Input	Expected	Got	
~	3	Case #1	Case #1	~
	3	10203010011012	10203010011012	
	4	**4050809	**4050809	
	5	****607	****607	
		Case #2	Case #2	
		1020304017018019020	1020304017018019020	
		**50607014015016	**50607014015016	
		****809012013	****809012013	
Ca 10 **		*****10011	*****10011	
		Case #3	Case #3	
		102030405026027028029030	102030405026027028029030	
		**6070809022023024025	**6070809022023024025	
		****10011012019020021	****10011012019020021	
	*****13014017018 ******15016		*****13014017018	
			******15016	

Passed all tests!

Finish review